In re Patent Application of:
LEVINE ET AL.
Serial No. **09/656,393**
Filing Date: **9/6/2000**
_____/

In the Claims:

1. (CURRENTLY AMENDED) A computer implemented method for managing mobile workers in an object oriented programming environment comprising the steps of:

classifying within a database of a computer a plurality of target objects corresponding to facilities assets to be worked on by a mobile worker;

defining attributes of each target object, including any tasks to be performed on each target object;

scheduling mobile workers for the tasks to be performed on target objects by running a rule engine to determine algorithms ~~and heuristics to be used to schedule mobile workers for the tasks to be performed~~ based on a utility function for partitioned jobs and workers wherein different algorithms are selected and used for different partitions to schedule jobs and workers in selected different regions;

outputting a schedule of jobs to the mobile workers, and further comprising the step of creating jobs as a collection of tasks for a target that is to be scheduled and controlled by a policy as the definitions, rules and business factors that control the behavior of system agents, comprising a planner agent that inventories items requiring work and determines tasks to schedule, the skills required to complete the tasks and material needs;

a schedule agent that matches skill resources to the demands of the job and creates a proposed schedule;

a dispatcher agent for tracking the location and status of the workforce;

a job state manager that maintains the state of active jobs and determines which jobs transition to new states;

an event bus operative with the system agents and database, wherein said system agents communicate across the event bus with the database and rule engine for implementing system agent functions based on events passed over the event bus; and

a simulator module and simulator database from which data has been copied from the database for target objects wherein the simulator module queries the simulator database for data to determine the effects of a policy change on planning and scheduling of jobs and workers using the different algorithms and partitions.

2. (ORIGINAL) A method according to Claim 1, and further comprising the step of classifying the plurality of target objects within a server computer and outputting the schedule to a client computer operated by a mobile worker.

3. (ORIGINAL) A method according to Claim 1, and further comprising the step of communicating with a mobile worker via a telecommunications link and a hand-held, web based device.

4. (ORIGINAL) A method according to Claim 1, and further comprising the step of building a plurality of user configured system agents for one of at least automating work planning, scheduling tasks to workers, dispatching workers, stores management, job state management or end-of-shift management.

5. (ORIGINAL) A method according to Claim 4, wherein the rule engine comprises a forward chaining rule engine with different rule sets for each system agent.

6. (ORIGINAL) A method according to Claim 1, wherein the rule engine determines a primary scheduling algorithm and parameters to be used for scheduling jobs to workers.

7. (PREVIOUSLY PRESENTED) A method according to Claim 6, wherein the primary scheduling algorithm comprises a brute force scheduling algorithm that is operable by determining an n number of jobs and m workers, trying all combinations of n jobs on the schedules for m workers and choosing assignments that maximize a total utility of the workers' schedules.

8. (ORIGINAL) A method according to Claim 6, wherein the primary scheduling algorithm comprises a round robin scheduling algorithm that is operable by assigning jobs to mobile workers sequentially after ordering an unassigned job queue based on a change in job utility.

9. (PREVIOUSLY PRESENTED) A method according to Claim 6, wherein the primary scheduling algorithm comprises a scheduling algorithm that assigns jobs to workers that maximize a job's utility.

10. (PREVIOUSLY PRESENTED) A method according to Claim 9, wherein an unassigned job queue is ordered from a highest utility to a lowest utility.

11. (PREVIOUSLY PRESENTED) A method according to Claim 9, wherein an unassigned job queue is ordered from a lowest utility to a highest utility.

12. (PREVIOUSLY PRESENTED) A method according to Claim 1, wherein the algorithm comprises a rescheduling algorithm that is operable by determining a utility of unassigned jobs and rescheduling assigned jobs, replacing some assigned jobs with unassigned jobs on workers' schedules, based on an added utility.

13. (ORIGINAL) A method according to Claim 1, and further comprising the step of maintaining a historical database that reflects all changes in system configuration, including targets and tasks, based on running system agents and on user interactions.

14. (ORIGINAL) A method according to Claim 1, and further comprising the step of viewing status and changes of task, system agents and schedules of jobs within a business viewer.

15. (ORIGINAL) A method according to Claim 1, and further comprising the step of maintaining a system log of all activities.

16. (ORIGINAL) A method according to Claim 1, and further comprising the step of maintaining a policy database that allows users to configure system agents and a plurality

of use cases corresponding to human and system interaction and definitions.

17. (ORIGINAL) A method according to Claim 1, and further comprising the step of building definitions of targets and their tasks, according to the classification of the targets as templates, and using the templates to create each individual target of the classification.

18. (CURRENTLY AMENDED) A computer implemented method for managing mobile workers in an object oriented programming environment comprising the steps of:

classifying attributes of each target object, including the tasks to be performed on each target object;

building user configured system agents and software components that automate the system environment for managing mobile workers;

scheduling mobile workers for tasks to be performed on target objects by running a rule engine to determine algorithms ~~and heuristics to be used to schedule mobile workers for the tasks to be performed~~ based on a utility function for partitioned jobs and workers wherein different algorithms are selected and used for different partitions to schedule jobs and workers in selected different regions;

configuring system agents and software components with user configured settings of a policy database that are reflective of a particular business;

outputting a schedule of jobs to the mobile workers, and further comprising the step of creating jobs as a collection of tasks for a target that is to be scheduled and controlled

by a policy as the definitions, rules and business factors that control the behavior of system agents, comprising a planner agent that inventories items requiring work and determines tasks to schedule, the skills required to complete the tasks and material needs;

a schedule agent that matches skill resources to the demands of the job and creates a proposed schedule;

a dispatcher agent for tracking the location and status of the workforce;

a job state manager that maintains the state of active jobs and determines which jobs transition to new states;

an event bus operative with the system agents and database, wherein said system agents communicate across the event bus with the database and rule engine for implementing system agent functions based on events passed over the event bus; and

a simulator module and simulator database from which data has been copied from the database for target objects wherein the simulator module queries the simulator database for data to determine the effects of a policy change on planning and scheduling of jobs and workers using the different algorithms and partitions.

19. (PREVIOUSLY PRESENTED) A method according to Claim 18, and further comprising the step of updating the policy database interactively wherein the system agents and other software components update their actions based on any present contents of the policy database.

20. (PREVIOUSLY PRESENTED) A method according to Claim 18, and further comprising the step of simulating any workings of the system environment for determining any values to be used within the policy database.

21. (PREVIOUSLY PRESENTED) A method according to Claim 20, wherein said step of simulating further comprises the step of:

a) setting policy database values;

b) simulating resultant operations of system agents and software components and viewing the results;

c) iterating between steps a and b to view the impact of setting policy database variables to various values; and

d) using the results of a through c to determine any optimum values to use for the policy values in a live operational system.

Claims 22-25 (CANCELLED)

26. (CURRENTLY AMENDED) A computer implemented method for managing mobile workers in an object oriented programming environment comprising the steps of:

classifying within a database of a computer a plurality of target objects corresponding to facilities assets to be worked on by a mobile worker;

defining any attributes of each target object, including any tasks to be performed on each target object;

scheduling mobile workers for the tasks to be performed on target objects by running a rule engine to determine any algorithms ~~and heuristics to be used to schedule mobile~~

~~workers for the tasks to be performed~~ based on a utility function for partitioned jobs and workers wherein different algorithms are selected and used for different partitions to schedule jobs and workers in selected different regions;

and further comprising the step of creating jobs as a collection of tasks for a target that is to be scheduled and controlled by a policy as the definitions, rules and business factors that control the behavior of system agents, comprising a planner agent that inventories items requiring work and determines tasks to schedule, the skills required to complete the tasks and material needs;

a schedule agent that matches skill resources to the demands of the job and creates a proposed schedule;

a dispatcher agent for tracking the location and status of the workforce;

a job state manager that maintains the state of active jobs and determines which jobs transition to new states;

an event bus operative with the system agents and database, wherein said system agents communicate across the event bus with the database and rule engine for implementing system agent functions based on events passed over the event bus;

establishing a simulator database based on data copied from the database of target objects and running a simulator ~~program~~ module to establish policy values in a simulation of a working of a system environment to determine ~~optimum policy values for a given business~~ a policy change on planning and scheduling of jobs and workers using the different algorithms and partitions.

27. (ORIGINAL) A method according to Claim 26, and further comprising the step of classifying the plurality of target objects within a server computer and outputting the schedule to a client computer operated by a mobile worker.

28. (ORIGINAL) A method according to Claim 26, and further comprising the step of communicating with a mobile worker via a telecommunications link and a hand-held, web based device.

29. (ORIGINAL) A method according to Claim 26, and further comprising the step of building a plurality of user configured system agents for one of at least automating work planning, scheduling tasks to workers, dispatching workers, stores management, job state management or end-of-shift management.

30. (ORIGINAL) A method according to Claim 29, wherein the rule engine comprises a forward chaining rule engine with different rule sets for each system agent.

31. (ORIGINAL) A method according to Claim 26, wherein the rule engine determines a primary scheduling algorithm and parameters to be used for scheduling jobs to workers.

32. (PREVIOUSLY PRESENTED) A method according to Claim 31, wherein the primary scheduling algorithm comprises a brute force scheduling algorithm that is operable by determining an n number of jobs and m workers, trying all combinations of n

jobs on the schedules for m workers and choosing assignments that maximize a total utility of the workers' schedules.

33. (ORIGINAL) A method according to Claim 31, wherein the primary scheduling algorithm comprises a round robin scheduling algorithm that is operable by assigning jobs to mobile workers sequentially after ordering an unassigned job queue based on a change in job utility.

34. (PREVIOUSLY PRESENTED) A method according to Claim 31, wherein the primary scheduling algorithm comprises a scheduling algorithm that assigns jobs to workers that maximize a job's utility.

35. (PREVIOUSLY PRESENTED) A method according to Claim 34, wherein an unassigned job queue is ordered from a highest utility to a lowest utility.

36. (PREVIOUSLY PRESENTED) A method according to Claim 34, wherein an unassigned job queue is ordered a lowest utility to a highest utility.

37. (PREVIOUSLY PRESENTED) A method according to Claim 26, wherein the algorithm comprises a rescheduling algorithm that is operable by determining utility of unassigned jobs and rescheduling assigned jobs, replacing some assigned jobs with unassigned jobs on workers' schedules, based on an added utility.

38. (ORIGINAL) A method according to Claim 26, and further comprising the step of maintaining a historical database that reflects all changes in system configuration, including targets and tasks, based on running system agents and on user interactions.

39. (ORIGINAL) A method according to Claim 26, and further comprising the step of viewing status and changes of task, system agents and schedules of jobs within a business viewer.

40. (ORIGINAL) A method according to Claim 26, and further comprising the step of maintaining a system log of all activities.

41. (ORIGINAL) A method according to Claim 26, and further comprising the step of maintaining a policy database that allows users to configure system agents and a plurality of use cases corresponding to human and system interaction and definitions.

42. (PREVIOUSLY PRESENTED) A method according to Claim 36, and further comprising the step of building definitions of targets and their tasks, according to the classification of the targets as templates, and using the templates to create each individual target of the classification.

43. (CURRENTLY AMENDED) A system for managing mobile workers comprising:

a plurality of target objects classified within a database of a computer corresponding to facilities assets to be worked on by a mobile worker, each target object having defined attributes, including any tasks to be performed on each target object; and

a rule engine contained within the computer that is operable to determine any algorithms ~~and heuristics to be used to schedule mobile workers for the tasks to be performed~~ based on a utility function for partitioned jobs and workers wherein different algorithms are selected and used for different partitions to schedule jobs and workers in selected different regions, and heuristics to be used to schedule mobile workers for the tasks to be performed, and further comprising jobs that are formed as a collection of tasks for a target that is to be scheduled and controlled by a policy as the definitions, rules and business factors, and system agents that are controlled by the policy, said system agents comprising

a planner agent that inventories items requiring work and determines tasks to schedule, the skills required to complete the tasks and material needs;

a schedule agent that matches skill resources to the demands of the job and creates a proposed schedule;

a dispatcher agent for tracking the location and status of the workforce;

a job state manager that maintains the state of active jobs and determines which jobs transition to new states;

an event bus operative with the system agents and database, wherein said system agents communicate across the event bus with the database and rule engine for implementing

system agent functions based on events passed over the event bus; and

a simulator module and simulator database from which data has been copied from the database for target objects wherein the simulator module queries the simulator database for data to determine the effects of a policy change on planning and scheduling of jobs and workers using the different algorithms and partitions.

44. (ORIGINAL) A system according to Claim 43, wherein the computer comprises a server computer, and further comprising a communications link established with mobile workers over which a schedule of jobs is output to client devices of the mobile workers.

45. (ORIGINAL) A system according to Claim 43, and further comprising a plurality of user agents defined within the computer for one of at least automating work planning, scheduling tasks to workers, dispatching tasks to workers, dispatching workers, stores management, job state management or end-of-shift management.

46. (PREVIOUSLY PRESENTED) A system according to Claim 43, wherein said algorithm comprises a brute force scheduling algorithm that is operable by determining an n number of jobs and m workers, trying all combinations of n jobs on the schedules for m workers and choosing assignments that maximize a total utility of the schedules for the workers.

47. (ORIGINAL) A system according to Claim 43, wherein the algorithm comprises a round robin scheduling algorithm that is operable by assigning jobs to mobile workers sequentially after ordering an unassigned job queue based on a change in job utility.

48. (ORIGINAL) A system according to Claim 43, and further comprising a policy database having rules and data for user configuring system agents and use cases corresponding to human and system interaction and definitions.

49. (ORIGINAL) A system according to Claim 43, and further comprising a simulator database and simulation program for establishing optimum policy values.